PTO:

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 09/557,633 | 04/25/2000 | Leona Dryden Baumgart | ST9-97-054 | 9263 |

7590 06/17/2003

David Victor Esq
315 South Beverly Drive
Suite 210
Beverly Hills, CA 90212

| EXAMINER |
|---|
| VU, TUAN A |

| ART UNIT | PAPER NUMBER |
|---|---|
| 2124 | 6 |

DATE MAILED: 06/17/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

| Office Action Summary | Application No. | Applicant(s) | |
|---|---|---|---|
| | 09/557,633 | BAUMGART ET AL. | |
| | Examiner | Art Unit | |
| | Tuan A Vu | 2124 | |

*-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --*

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on *04/17/03* .

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) *1-33* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-33* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *25 April 2000* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on _____ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

**Priority under 35 U.S.C. §§ 119 and 120**

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

**Attachment(s)**

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449) Paper No(s) *5* .
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other:

## DETAILED ACTION

1. This action is responsive to the response filed April 17, 2003.

Claims 1-33 have been submitted for examination.

### *Claim Rejections - 35 USC § 102*

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

3. Claims 1, 3, 5-7, 12, 14, 16-18, 23, 25, and 27-29 are rejected under 35 U.S.C. 102(b) as being anticipated by Lee, USPN: 5,553,286 (hereinafter Lee).

**As per claim 1**, Lee discloses a method, system (col. 4, lines 48-59; Fig. 1) of producing an executable file including:

receiving a plurality of programming language statements (e.g. col. 2, lines 3-7; Fig. 5, *TEXT2/TEXT* );

translating the source program (e.g. col. 1, lines 5-32; col. 4, lines 41-47; col. 5, line 58 to col. 6, line 6), including a symbol reference( *array of indexed data* – col. 6, lines 30-31); a symbol definition ( Fig. 4, *section, class name*); attribute information for the symbol reference ( *index numbers, size and offset fields* -- col. 6, line 51 to col. 7, line 1; *size, sequence, starting offset* – col. 8, lines 37-47), and attribute information for the symbol definition (*binding attribute* – col. 8, lines 37-46);

binding object modules into a program object, wherein the attribute information is available when so binding (e.g. col. 8, lines 37-54; col. 7, lines 4-30 )

resolving an external symbol reference ( col. 8, lines 6-33; Fig. 5).

**As per claim 3**, Lee discloses that the object module is further capable of including fixed attribute information (e.g. RMODE - col. 7, lines 4-30, 47-49; *length, location* -- col. 9, lines 38-45; col. 3, lines 14-15).

**As per claim 5**, Lee discloses an address constant (*adcons* – col. 1, lines 48-60) for a symbol (Fig. 4, *section, class name)* and the attribute information declaring attribute information for the address constant (*class offsets, class identifier* -- col. 8, lines 48-52; *location and type of each address constant* - col. 3, lines 16-18).

**As per claim 6**, Lee discloses additional address constants for additional references to the symbol reference in the object module (*one or more address constants* -- col. 1, lines 56-61; col. 7, lines 41-45) and different attribute information sets for the address constants(e.g. col. 8, lines 50-54; *target segment, offsets, virtual address* -- col. 9, lines 6-14, 28-35, 54-67; col. 7, lines 1-3).

**As per claim 7**, Lee discloses that the resolving of the symbol reference and definition comprises a compatibility check *(Binder* ) using the attribute information (*binding attribute, class identifier, offset* -- col. 8, lines 6-54; Fig. 5); and a separate compatibility checking for each reference to a symbol ( col.6, line 51 to col. 7, line 1) for which there is a separate address constant and separate attribute information for each address constant ( *target segment, offsets, virtual address* -- col. 9, lines 6-14, 28-35, 54-67).

**As per claims 12, 14 and 16**, these claims recite a system comprising the same corresponding limitations set forth in claims 1, 2 and 5 above, respectively. Hence, the rejections of claims 1, 2, and 5 are herein applied.

**As per claims 17 and 18**, these are the system versions of the methods in respectively, claims 6 and 7 above; whose limitations have already been addressed above.

**As per claim 23, 25 and 27**, these claims recite an article of manufacturer comprising a computer usable media embedding a computer program capable of performing the same method steps in claims 1, 2, and 5 respectively, which limitations have been addressed in the rejection of claims 1, 2, and 5, respectively. Furthermore, Lee also teaches such computer product/article of manufacturer in col. 10, line 66 to col. 12, line 9.

**As per claims 28 and 29**, these are the computer product versions of the methods in respectively, claims 6 and 7 above; whose limitations have already been addressed above.

***Claim Rejections - 35 USC § 103***

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 2, 4, 8, 9; 13, 15, 19, 20; and 24, 26, 30, 31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lee, USPN: 5,553,286, as applied to claims 1, 12, and 23 above, and further in view of Fitzgerald, USPN: 5,408,665 (hereinafter Fitzgerald).

**As per claim 2**, Lee does not explicitly disclose that the language statement is capable of indirectly declaring extended attribute information defined in another location in the object module. Lee, however, discloses external symbol storage and relocation dictionary ( col. 2, lines 1-9). Further, Fitzgerald, in a method to bind object modules translated from program source into executables using external symbol resolution analogous to that of Lee, discloses the use of

extended Dictionary (e.g. col. 10, line 47 to col. 11, lines 35) and extended attribute information indirectly defined in another location of the object module (*Module ID* – Fig. 4b, 5b, 6B,C, D; – Note: the attribute MODULE ID is set to link via a pointer in the external dictionary to get to an attribute BLOCK ID declared externally to the module in which MODULE ID has been referenced ; and this is equivalent to including extended attribute to reference to another attribute declared externally, indirectly via a pointer). It would have been obvious for one of ordinary skill in the art at the time the invention was made to implement the indirect definition of an attribute via the use of extended attribute information as taught by Fitzgerald to the method of translating source program into object modules using external symbol resolution disclosed by Lee; and if the program statement does not already include such indirect referencing via the extended attribute, then to provide program statements to include such attribute so as to be able to refer to it to access another attribute declared externally as shown by Fitzgerald. The modification would have been obvious because this additional source of extended attribute information made ready into structures at binding time would facilitate the relocating of internal and external symbol references via indirect referencing only, thus improve the time and direct storage resource usage efficiency of the modules linking and binding process in Lee's disclosed system, such storage efficiency being evidenced by use of pointer just as suggested by Fitzgerald to alleviate cumbersome overloading of data reference at one address location.

**As per claim 4**, this claim is rejected for the same reasons set forth in claim 2 above.

**As per claim 8**, Lee further discloses that the object module includes an External Symbol Directory (ESD) including a record capable of indicating a symbol in the program, a location of the symbol in the program (col. 6, lines 54-57; Fig. 5, *ESD 64*); but does not point out that such

ESD includes a pointer to attribute information in the program for the symbol. However, Fitzgerald in a method to bind object modules translated from program source into executables analogous to that of Lee discloses the use of pointer in an external symbol table (*External Dictionary*) to refer to the attribute information in the program for the symbol (e.g. Fig. 4b, 5b, 6B,C, D). It would have been obvious for one of ordinary skill in the art at the time the invention was made to add the implementation of pointer to symbol attribute inside the ESD as taught by Fitzgerald into Lee's ESD because such additional pointing structure would facilitate the fetching of attribute needed to resolve symbol references during binding of modules components, and alleviate the data (e.g. whole attribute data) store usage in the ESD by just storing an address referring to that data.

**As per claim 9**, Lee discloses that the object module further includes an Relocation List Directory (RLD), a location of an address constant (col. 3, lines 16-18; col. 8, lines 48-54; Fig. 5 – *RLD 64*); but does not point out that such RLD includes a pointer to attribute information for the address constant in the program. However, Fitzgerald in a method to bind object modules translated from program source into executables analogous to that of Lee discloses the use of external tables (e.g. step *603*-Fig. 6A; *external list* – Fig. 5B); pointer in a symbol dictionary (*STD. DICT.*) and External Dictionary to point to the second attribute information (e.g. in form of an identification string) indirectly defining the first referenced program module attribute (Fig. 4b, 5b, 6B,C, D; *MODULE ID* → *PTR* (*Ext. Dict*) → *ASCII STRING, BUCKET*- Fig. 5B). It would have been obvious for one of ordinary skill in the art at the time the invention was made to add the implementation of pointer to address constant inside an external table or symbol directory, i.e. relocating tables, as taught by Fitzgerald into Lee's RLD because such additional

pointing structure would facilitate the fetching of attribute needed to resolve symbol references during binding of modules components, and alleviate the data (e.g. whole attribute data) store usage in the ESD by just storing an address referring to that data.

**As per claims 13, 19, 20, and 24, 30, 31**,these claims are respectively the system and computer product versions of the method limitations set forth and addressed respectively, in claims 2, 8, and 9 above. Hence, the rejections of claims 2, 8 and 9, respectively, are herein applied.

**As per claims 15 and 26**, in reference to respectively, claims 12 and 23 above, these are respectively the system and computer product versions of the method in claim 4 above. Hence, the rejections of claim 4 are herein applied for both instant claims.

6. Claims 10-11, 21-22, and 32-33 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lee, USPN: 5,553,286, as applied to respectively, claims 1, 12, and 23 above, and further in view of Looney, USPN: 6,366,876 (hereinafter Looney).

**As per claim 10**, Lee discloses that the resolving of the symbol reference and definition comprises a compatibility check (see claim 7 above); but does not explicitly disclose that such resolving further comprises a compatibility check using signature data for the symbol definition and symbol reference. However, Looney in a method of assessing compatibility between programming resources analogous to that of Lee discloses the use of signature data per symbol reference (*MethodRef* – col. 12, lines 1-5) and symbol definition in the compatibility check within the symbol reference/definition resolution process (*signature attribute for method* -- col. 10, lines 3-42). It would have been obvious for one of ordinary skill in the art at the time the invention was made to implement the signature data for symbol reference and definition just as

taught by Looney into the method of compatibility check disclosed by Lee because this would further enforce the compliance of referenced data to be matched during the binding/linking process by virtue of the pre-determined signature data; thereby obviating extraneous usage of resources for recovery due to incompatibility errors.

**As per claim 11**, Lee does not expressly disclose the step of determining whether the compatibility check failed and the step of processing the attribute information declared for the symbol reference and definition that failed the compatibility check to determine a cause of the incompatibility. But Looney in an analogous method discloses the determining as to whether the symbol reference and definition compatibility check fail (Fig. 9a,b,c ); and the processing of the attribute information (e.g. *compliance status* – col. 10, lines 20-64*)* for such symbol reference and definition( *class, method, fields, return type*) to determine the cause of incompatibility(e.g. col. 2, lines 19-47; col. 2 line 59 to col. 3, line 6; col. 10, lines 20-64; col. 12, lines 54-58). It would have been obvious for one of ordinary skill in the art at the time the invention was made to add the compatibility failure checking for symbol reference and definition; and determining of cause thereof such as taught by Looney into the method of compatibility check disclosed by Lee. The modification would have been obvious because this would further establish a systematic recording of the compliance checking results on referenced data during the binding/linking process, thereby providing a base of information for the analysis and/or improvement of future incompatibility error checking processes.

**As per claims 21, 22 and 32, 33**, these are respectively the system and computer product versions of the methods in respectively, claims 10 and 11 above. Hence, the rejections of claims 10 and 11, respectively, are herein applied.

***Response to Arguments***

7. Applicant's arguments filed 4/17/2003 have been fully considered but they are not persuasive. Following are the reasons therefor.

**As per claims 1, 3, 5-7, 12, 14, 16-18, 23, 25, and 27-29** being rejected under 35 U.S.C. 102(b) over Lee, USPN: 5,553,286:

**As per claims 1, 12, and 23,**

(A) Applicants traverse Examiner's cited portions of Lee because "the cited index, size, offset information concern limitations of a program object ... available" and "the cited index, size ... structural information that defines the layout of the program object"( Applicant's remarks, p.6, 2nd paragraph).

In response, Examiner notes that the index numbers being widened as cited in lines 51-61 do not necessarily mean they are not attribute associated with symbols or references to sections or modules as described. They are increased to obviate spatial restrictions issues but do represent information attributed to those sections of modules as much as the size and offsets fields which are also widened to provide for restrictions in memory. But they all are still fields or attribute information supporting the referenced symbol, pointers or sections names as mentioned in the cited portions. The claimed limitation states that the object module is capable of including ... "attribute information for the symbol reference" (*pointers*), ... symbol definition ( *external names, sections names, class names*) derived from language statements". Size and offset can be attributed to inform on size of modules or referenced sections defined, hence they are attribute information to symbols defined and referred to by the program. Index numbers are attributes because they inform on the location of certain referred to elements within the object program, as

normally understood by one skilled in the art. Further, they are all included in the program object (e.g. Fig. 4), and contributed to inform on location of sections or names referred to by the binding process, or load module process. Besides, the claim as recited does not exclude the fact that the attribute information is or can be for defining or accommodating program object layout. The cited portion seems to address the memory limitations but the presence of all the above attributes as to inform on referred parts of the module to bind are disclosed and Lee has fulfilled the limitation as claimed.

(B) Applicants have asserted that "the cited col. 8, lines 37-47 are used by the binder...create a single program object" and that " index, size, and offset information ... concern limitations of the program.." and that "col. 8 discusses how the binder ... overlay items when creating the single program object"(Applicant's remarks, p.6, paragraph 3 to p. 7, 1st 2 lines). In response, Examiner notes that the argument bear the same rationale, i.e. layout of program versus including attribute information, as mentioned on section (A) above, hence will refer to Examiner's response in section A to address the "overlay" feature as mentioned by the Applicants, because although the attributes are used in defining the layout of the module, they do not fail to read into the claimed limitation as has been pointed out in section A above.

**As per claims 3, 14, and 25,**

(C ) Applicants have asserted that " the cited col. 7, lines 4-30 discuss class attribute .. binding variations to a finite set of class attributes" and that "Although ... for the symbol reference and definition as claimed" (Applicant's remarks, p.7, paragraphs 6, 7). In response, Examiner notes that length, location as cited are fixed attributes for referencing module to load from the DASD as much as RMODE is a fixed attribute for the location of a resident load module from the

DASD (see col. 3, lines 3-44; col. 7, lines 15-16). Further, even the specification of the application itself specifies that RMODE can be one of the properties for fixed attributes such as residency mode (specifications, p. 11, ll. 24-25). The attributes as cited in the rejection are included in the program object and fulfill the limitation that they are attribute information for the symbol reference and definition the same way as mentioned in section A above in the binding process by Lee.

(D ) Applicants have asserted that " the loader opens a file ... This cited section nowhere ... declaring attribute information for the symbol reference and definition" (Applicant's remarks, p.7, paragraph 8). In response, Examiner notes that Examiner has cited this portion to address the fact that the loading process utilizes the fixed attributes *length, location* to retrieve resident load module from the DASD, for which the RMODE attribute as mentioned above could play a role. Therefore, Examiner's rationale for addressing Applicants' assertion that this cited portion is not disclosing the fixed attribute information is same as the one used in section (C ) above.

**As per claims 5, 16, 27,**

(E) Applicants have asserted that " col. 8 mentions that the binder stores the target address ... Nowhere does the cited col. 8 ... that an address constant provides attribute information for a symbol available during binding"( Applicant's remarks, p.8, paragraph 4). In return, Examiner would like to point out that the spatial attribute to inform of the location of an address constant has been disclosed in the specification of the instant application (p. 12, ll. 1-11; Figure 3, lines 120, 122); and that analogously, in view of Lee, the offset is stored inside the *adcon* and its class identifier is stored in the RLD, both being viewed as attribute information as well as the type and location of the address constant inside the RLD as disclosed by col. 3, lines 14-15 as cited in the

rejection. Hence, Lee has fulfilled the limitation as including an attribute information so as to locate the address constant as claimed.

( F) Applicants have asserted that claims 6, 7; 17-18; 28-29 are being rejected on the grounds of Lee's disclosing address constant but that no attribute information for symbol references has been provided by Lee (Applicant's remarks, p.8, last paragraph ); such argument is joined by the same rationale used by Applicants in arguing about claims 5, 16, 27 above. For that, Examiner will address such argument by using section (E ) rationale to point out that Lee has fulfilled the limitation of the address constant and its related attribute information.

**As per claims 2, 4, 8, 9; 13, 15, 19, 20; and 24, 26, 30, 31** being rejected under 35 U.S.C. 103(a) over Lee, USPN: 5,553,286, in view of Fitzgerald, USPN: 5,408,665:

**As per claims 2, 13, and 24,**

(G) Applicants have asserted that "the EXTDEF is an external ...public symbols in source and object modules... Nowhere in the cited Fitzgerald is there any ... object module" (Applicant's remarks, p.9, paragraph 4). The rejection now points to exactly why the cited Fitzgerald teachings fulfill the limitation that " the language statement is capable of indirectly declaring extended attribute information defined in another location in the object module". As per the present rejection, the cited figures start by referring to the attribute MODULE ID which via a pointer in the Extended Directory accesses another attribute declared external to where the reference has started. The final BLOCK ID or STR LENGTH in Fig. 4B, are attributes indirectly defined in an external module but are accessed via extended attributes such as MODULE ID from the referring module. And this paradigm is analogous to the example of "symbol BB" and attribute "Label" shown in the application specification on pg. 11, lines 11-23.

Examiner would like to emphasize on the fact that Fitzgerald suggests the use of pointer to indirectly refer to a external attribute defining the starting attribute, e.g. MODULE ID, referred within the calling module and that such teaching is the basis for the rationale used in the rejection as opposed to the misleading use of symbols like EXTDEF or EXTERN or col. 12, ll. 5-33 as previously cited in the first action rejection. That being said, the combination of Fitzgerald and Lee has fulfilled the limitation as claimed.

**As per claims 8, 19, and 30,**

(H) Applicants have asserted that "the cited col. 15 discusses a pointer ... for symbol definitions and references as claimed"(Applicant's remarks, p.10, paragraph 2). In response, the discussion using a pointer has now been evidenced by more figures in the rejection and more clearly addresses the use of table by Fitzgerald to relocate a referenced symbol by having a pointer to attribute information for symbol, an attribute or the address constant such as suggested by Fitzgerald. The previous use of col. 15 to demonstrate the use of pointer to refer to symbols during the relocation process still has its teaching in that Fitzgerald does use a pointer in a extended directory to direct the binding process to get to the attribute or symbol string defined outside the calling module; but the present rejection is more illustrative as to how Fitzgerald teaches a pointer to the attribute as claimed. Further, Applicants have asserted that "Nowhere does the cited col. 15 of Fitzgerald ... pointer is in the RLD" (Applicant's remarks, p.10, last paragraph). The use of external tables, dictionary and extended dictionary by Fitzgerald are analogous to the use of RLD by Lee, therefore the combining of both teachings as expressed in the rejection has fulfilled the limitation for the reason set forth therein. The issue raised by col. 15 only addresses part of the limitation as has been mentioned above; and the now used reference

in the rejection should help clarify that by virtue of combining the address constant and RLD by Lee and tables and pointers by Fitzgerald, with both Lee and Fitzgerald having analogous purposes in binding object modules, the limitations in claim 9 are set as obvious as in the rejection.

***Conclusion***

8. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (703)305-7207. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (703)305-9662.

**Any response to this action should be mailed to**:

Commissioner of Patents and Trademarks

Washington, D.C. 20231

**or faxed to:**

(703) 746-7239, ( for formal communications intended for entry)

**or:** (703) 746-7240 ( for informal or draft communications, please label "PROPOSED" or "DRAFT")

Hand-delivered responses should be brought to Crystal Park II, 2121 Crystal Drive, Arlington. VA. , 22202. 4th Floor( Receptionist).

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

VAT
June 6, 2003

KAKALI CHAKI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100